Building a Conversational AI Assistant for African Travel Services with LLMs and RAG

Grace Kevine Tadaha Ngoufo[1]
Shamsuddeen Hassan Muhammad[2], **Kevin Jeff Fogang Fokoa**[1]

[1]AIMS Cameroon, [2]Imperial College London
Correspondence: kevine.tadaha@aims-cameroon.org, s.muhammad@imperial.ac.uk

Abstract

Travel agencies in many African countries face increasing pressure to handle large volumes of customer inquiries with limited staff or, either non-existent or outdated rule-based chatbots. To address this challenge, we develop a conversational virtual assistant powered by a Large Language Model (LLM) and enhanced with a Retrieval-Augmented Generation (RAG) pipeline. The system combines LLM reasoning, company-specific knowledge retrieval, and real-time API (Application Programming Interface) integration to deliver accurate, context-aware responses through WhatsApp, the region's most widely used communication platform. A dedicated web interface enables staff to upload and update internal documents, ensuring that the assistant remains aligned with changing service information. Demonstrations show that the proposed solution improves response speed, enhances user experience, and reduces operational burden.

1 Introduction

The rapid evolution of artificial intelligence worldwide has led to the emergence of Large Language Models (LLMs), which demonstrate exceptional abilities in natural language understanding, coherent text generation, and task automation. These models are becoming essential tools in various domains, offering new opportunities for intelligent decision-making and human–machine interaction. However, despite this global progress, the effective adoption of artificial intelligence (AI) technologies in many African countries remains limited. Several sectors continue to show reluctance toward integrating and trusting AI systems, even though such tools could significantly simplify operational workflows and contribute to economic growth and digital transformation across the continent (Azaroual, 2024).

In African contexts, and across varying levels of digital innovation and automation adoption, AI tools show an adoption rate of 41%, primarily focused on analytics, fraud detection and chatbots King et al., 2025. Due to restricted access to advanced technologies and infrastructure, as well as regulatory and policy gaps such as outdated regulations and insufficient data King et al., 2025, chatbots struggle to support natural and flexible conversations, often failing to deliver a satisfactory user experience. As a result, their adoption remains low, and organizations face increasing pressure to handle customer requests manually leading to service delays and operational inefficiencies.

These challenges are particularly visible in the travel industry. In this study, we focus on a travel agency offering a broad range of services, including flight booking, visa assistance, tourism packages, car rentals, and accommodation arrangements. Due to the large volume of client inquiries and limited human resources, the company often struggles to respond promptly and accurately to customer needs. This situation highlights the need for an adaptive and intelligent virtual assistant capable of managing diverse customer queries in real time while reducing the workload on agency staff.

In this paper, we address these challenges through two main system components. (1) We transform an existing rule-based chat-bot into a fully conversational AI assistant powered by a Retrieval-Augmented Generation (RAG) architecture, enabling the system to produce accurate, context-aware, and up-to-date responses grounded in the agency's verified documents. (2) We develop a dynamic and secure web platform that allows authorized staff to directly upload and update information (text or PDF form) in the chat-bot's knowledge database, ensuring that the system remains up to date, reliable, and aligned with changes in travel policies or service offerings.

In general, this work provides a practical demonstration of how LLM-based conversational systems can be effectively integrated into African service

environments, offering a foundation for future extensions such as multilingual support and deployment in other sectors.

2 Related Work

Conversational agents have been extensively studied in artificial intelligence, progressing from early rule-based systems to modern neural and LLM-driven architectures. Traditional dialog systems relied on hand-crafted rules and slot-filling approaches (Weizenbaum, 1966; McTear, 2002), which limited their flexibility and naturalness. The emergence of sequence-to-sequence models and transformer-based architectures (Vaswani et al., 2017) enabled more fluent responses, but these systems still required large task-specific datasets and often struggled to generalize beyond their training domain.

Recent advances in Large Language Models (LLMs), such as GPT-3 (Brown et al., 2020), PaLM (Chowdhery et al., 1), Llama-2/3 (Touvron et al., 2023) and Mistral (Jiang et al., 2024), have significantly improved conversational quality by leveraging large-scale pretraining. These models possess strong zero-shot and few-shot capabilities, enabling them to support open-domain and task-oriented dialogue with minimal supervision. However, LLMs are also prone to hallucinations (Maynez et al., 2020) and often produce incorrect or unverifiable information, making them unreliable for domains requiring precise factual knowledge such as travel regulations, visa requirements, and service policies.

To mitigate hallucinations, Retrieval-Augmented Generation (RAG) approaches combine LLMs with external knowledge sources (Lewis et al., 2020; Izacard and Grave, 2021). Retrieval-based augmentation has proven effective for grounding model outputs, improving factual accuracy, and ensuring up-to-date responses in dynamic domains, including customer service and information retrieval pipelines.

Within the African Natural Language Processing (NLPs) research landscape, significant progress has been made in addressing the scarcity of digital resources, datasets, and language technologies for African languages. Initiatives such as Masakhane (Nekoto et al., 2020), the AfriBERTa model (Ogueji et al., 2021), and the MasakhaNER 2.0 Africa-centric transfer learning or Named Entity Recognition (Adelani et al., 2022) efforts have contributed to multilingual NLP resources for low-resource African languages. Nevertheless, conversational AI applications tailored to African service industries remain under-explored. Existing chatbots deployed in African contexts often rely on rule-based or template-driven designs (Marone and Mbengue, 2025), leading to rigid interactions and limited scalability.

Despite the growing availability of African NLP resources, very few studies focus on LLM-driven conversational assistants for industry-specific workflows, such as travel agencies, tourism or customer support. Previous work on domain-specific assistants focuses mainly on general-purpose RAG pipelines (Gao et al., 2023) or enterprise knowledge systems, but does not address the unique infrastructure, data availability, or adoption challenges faced in African markets.

Our work contributes to filling this gap by demonstrating a practical use case of LLM-driven, RAG-based conversational assistance in an African travel agency context. Unlike previous rule-based systems used locally, we build a system capable of natural conversation, grounded retrieval, and dynamic knowledge updates through a dedicated web platform.

3 System Architecture

This section presents the architecture of the proposed conversational chat-bot, designed to improve the existing system with a more adaptive and scalable approach. The system integrates an LLM for natural and fluid conversation, a RAG pipeline using a vector database and an embedding model to provide to the bot precision on company-specific knowledge, external API (Application Programming Interface) for real-time information such as flight options, and Meta Webhook integration to enable conversations via WhatsApp.

3.1 Data Sources and Knowledge Database

To ensure the delivery of accurate and reliable information over time, our chat-bot relies on three primary data sources.

General Knowledge of the LLM One of the core sources of information used by our chat-bot is the general knowledge embedded in our LLM. Large language models are trained on vast and diverse corpora, including publicly available web documents, code, images, audio, video, and more; giving them broad world knowledge, linguistic patterns, and strong reasoning capabilities (Google

Figure 1: **Initial Login Interface:** The administrator accesses the system by entering authentication credentials, including a username and password, in order to securely log in and manage system functionalities.

DeepMind, 2024a). This general-purpose knowledge forms the foundational layer of our system and enables the model to provide coherent, contextually relevant responses even when domain-specific information is not available externally.

In this study, we use Gemini-2.5-flash, accessed through its official API, as the primary LLM powering the chat-bot. Gemini-2.5-flash is part of the Gemini family, developed by Google Deep-mind. It is designed for efficient, real-time inference, making it well-suited for interactive applications such as chat-bots (Google DeepMind, 2024b). The use of langchain community provide to the system an internal memory which help to keep the history of the conversation per user (Community, 2023).

To adapt the general knowledge of the LLM to the specific context of our application, namely the company domain, internal data, and user needs, we rely on prompt engineering. Prompt engineering uses carefully designed instructions, context templates, and query formulations to guide the LLM toward producing outputs aligned with the desired domain and style. This method has been widely studied and shown to significantly enhance the performance of LLMs across many tasks without changing their internal weights (Li et al., 2023).

External API (Application Programming Interface) An API is a set of rules and specifications that enables different software systems to communicate and exchange data and functionalities (Postman, 2024). In our work, we use several APIs provided by the company, particularly for retrieving IATA (International Air Transport Association) airport codes, fetching real-time flight availability based on the destination and flight type (round trip or one-way), and other operational information. The integration of these APIs with the LLM is achieved through the Model Context Protocol (MCP), an architecture primarily composed of a client–server interaction model that facilitates secure and efficient communication between the chatbot and external services (Protocol, 2023).

Specific Knowledge of the Company Knowing that LLMs only provide general knowledge, it is essential to supply them with company-specific information such as organizational activities, available products, and up-to-date service details. To address this need, we developed a dedicated web interface for staff members (Figure 2). Access to this platform is protected by an authentication mechanism (Figure 1), ensuring that only verified employees can upload, update, or manage internal company documents.

Through this secure interface, employees can upload or update internal documents, including text files and PDFs, containing information relevant to the company's services. Once uploaded, the documents are automatically processed, embedded, and stored in a vector database. This controlled access helps protect sensitive company information and prevents unauthorized disclosure or manipulation of internal data, while ensuring that the chatbot consistently provides accurate and up-to-date responses during user interactions.

In addition, the system supports document management functionalities, including the deletion of outdated or irrelevant documents and the computation of statistics to track the volume of documents available in the database.These features further contribute to data governance and privacy by allowing

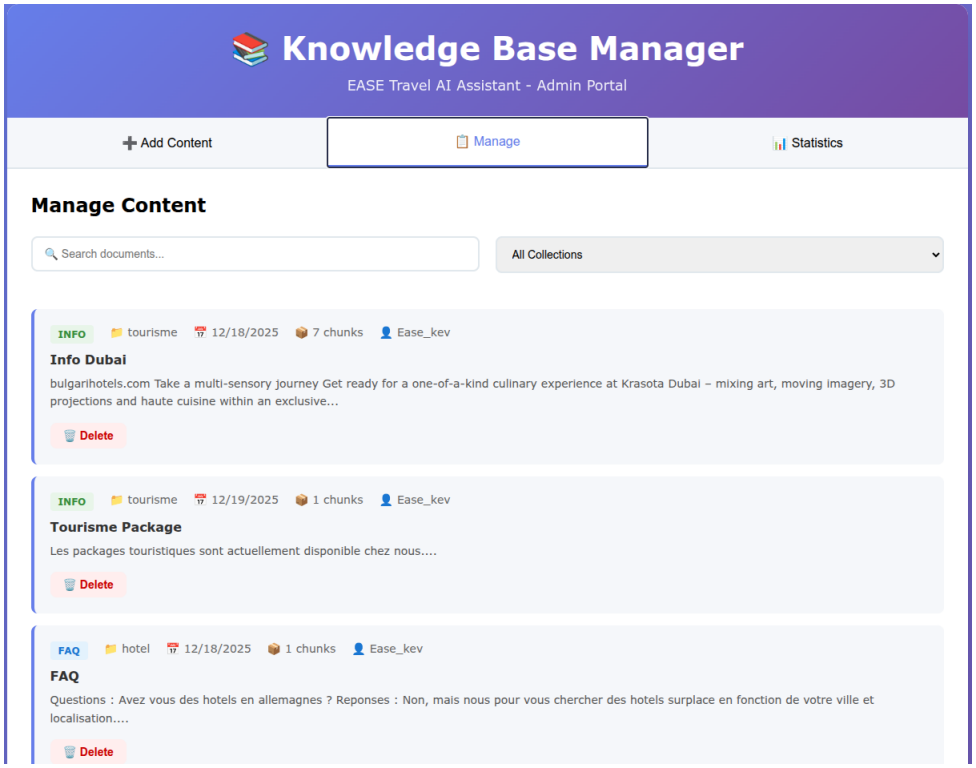

Figure 2: **Web Knowledge Base Manager:** Web-based interface for data management, allowing authorized staff to upload content, manage existing data, and view vector database statistics.

administrators to maintain only relevant and current information within the system.

To support this retrieval process, we rely on a vector database, a system that stores data in the form of high-dimensional numerical vectors. This structure enables efficient similarity search, allowing the chat-bot to retrieve the most relevant information based on semantic meaning rather than simple keyword matching. Several vector databases exist (e.g., Pinecone, Weaviate, Milvus), each with distinct capabilities. In this study, we employ Qdrant, an open-source vector database optimized for high-performance vector similarity search and scalable AI applications (Team, 2023). Qdrant was selected because it offers strong support for semantic search, delivers fast query performance, and remains free and user-friendly, making it well-adapted to our deployment constraints.

Before being stored in Qdrant, textual data must be transformed into vectors using an embedding model. This model converts text into numerical representations that preserve semantic meaning, allowing the system to compare and retrieve documents based on their conceptual similarity. For this purpose, we use the BGE-M3 embedding model, which supports dense and multi-vector retrieval as well as multilingual processing (Chen et al., 2024; Xiao et al., 2024). Our choice is motivated by its balance of accuracy and computational efficiency: the model is free, lightweight, and capable of generating rich embeddings that capture nuanced semantic relationships. These characteristics are essential for ensuring high-quality semantic search and, ultimately, improving the chat-bot's ability to deliver reliable, context-aware answers.

3.2 System Implementation

The full system implementation is designed around a hybrid architecture combining generation, retrieval, and real-time API interaction. Figure 3 shows an overview of the workflow, and the main components are described below.

User's Interface. To enable real-world deployment and effectively meet the operational needs of the travel agency, the chat-bot has been integrated with WhatsApp, which is the most widely used communication platform in the region. This choice ensures that users can interact with the system through a familiar and accessible interface, minimizing barriers to adoption.

The integration follows a structured workflow: When a user sends a message to the agency's WhatsApp Business number, the message is first forwarded by the Meta Developer Webhook to the backend server. The backend then processes the query, leveraging the chat-bot's natural language understanding and retrieval capabilities to generate

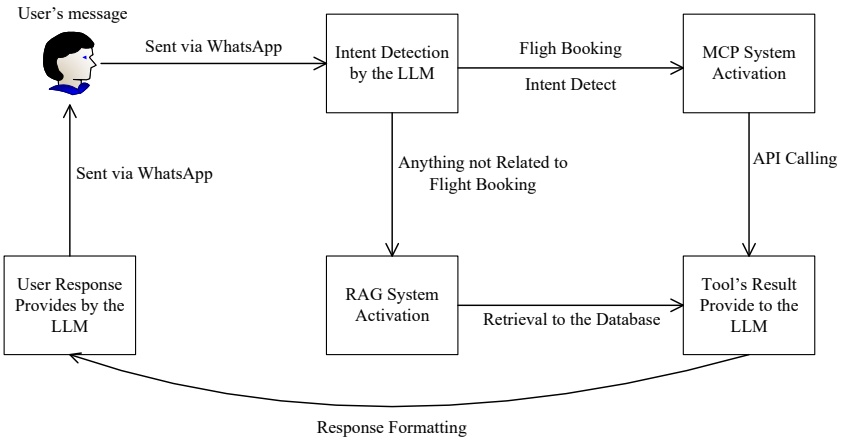

Figure 3: **System Architecture:** Our hybrid conversational assistant workflow combining intent-based routing, Multi-Component Processing (MCP), and Retrieval-Augmented Generation (RAG). User messages received via WhatsApp are first classified by the LLM; flight booking intents trigger API calls, while general travel queries activate the RAG system to retrieve knowledge from the database.

an appropriate response. Once generated, the response is transmitted back to the user through the WhatsApp Cloud API, completing the interaction cycle. This architecture not only guarantees real-time communication but also ensures scalability, reliability, and ease of maintenance. By leveraging a widely adopted messaging platform, the system provides a seamless and intuitive user experience while supporting the agency's operational objectives and enhancing customer engagement.

Backend Service Once a user message is captured and forwarded to the backend server, it is first processed by the Gemini model, which is responsible for interpreting the message and performing intent detection. If the query is general in nature, for instance, greetings or requests for broad, non-specific information, the model responds directly using its pre-trained knowledge. However, if the query pertains to the company's services, the system activates one of two specialized orchestration pathways.

For flight-ticket booking intents, the Model Context Protocol (MCP) system is triggered. In this case, the external flight-booking API is not invoked immediately. Instead, the Gemini model initiates a clarification dialogue to gather all essential booking parameters from the user, including destination, number of travellers, travel dates, and flight type. Once the information is complete, the MCP system retrieves the corresponding results. These results are then passed back through the Gemini model, which reformats, verifies, or restructures the output as needed before producing the final user-ready response.

For all other service-related intents not involving flight booking, the system activates a Retrieval-Augmented Generation (RAG) pipeline. In this workflow, the embedding of the user query is compared against stored document embeddings within the company's vector database. The most relevant information is retrieved (result from the RAG retrieval) and supplied to the Gemini model alongside the original query. The model then integrates this domain-specific context with its generative capabilities to produce a precise, consistent, and contextually grounded response.

The entire orchestration is governed by carefully engineered prompts that direct the model's behaviour across different scenarios. This design is further strengthened by few-shot prompting strategies, which provide the model with concrete examples of expected behaviour, thereby enhancing its ability to generalize across diverse queries while ensuring accuracy, consistency, and alignment with the company's operational requirements.

4 Usability and Analysis

Travel companies often face a high volume of customer inquiries while lacking sufficient customer service staff, leading to long waiting times and customer dissatisfaction. Existing procedural chatbots are limited by rigid, step-by-step interactions that can be slow and cumbersome. To address this issue, we propose a conversational chatbot that allows customers to interact naturally via WhatsApp and receive immediate, relevant responses without human intervention. The system was developed and deployed in collaboration with a real travel

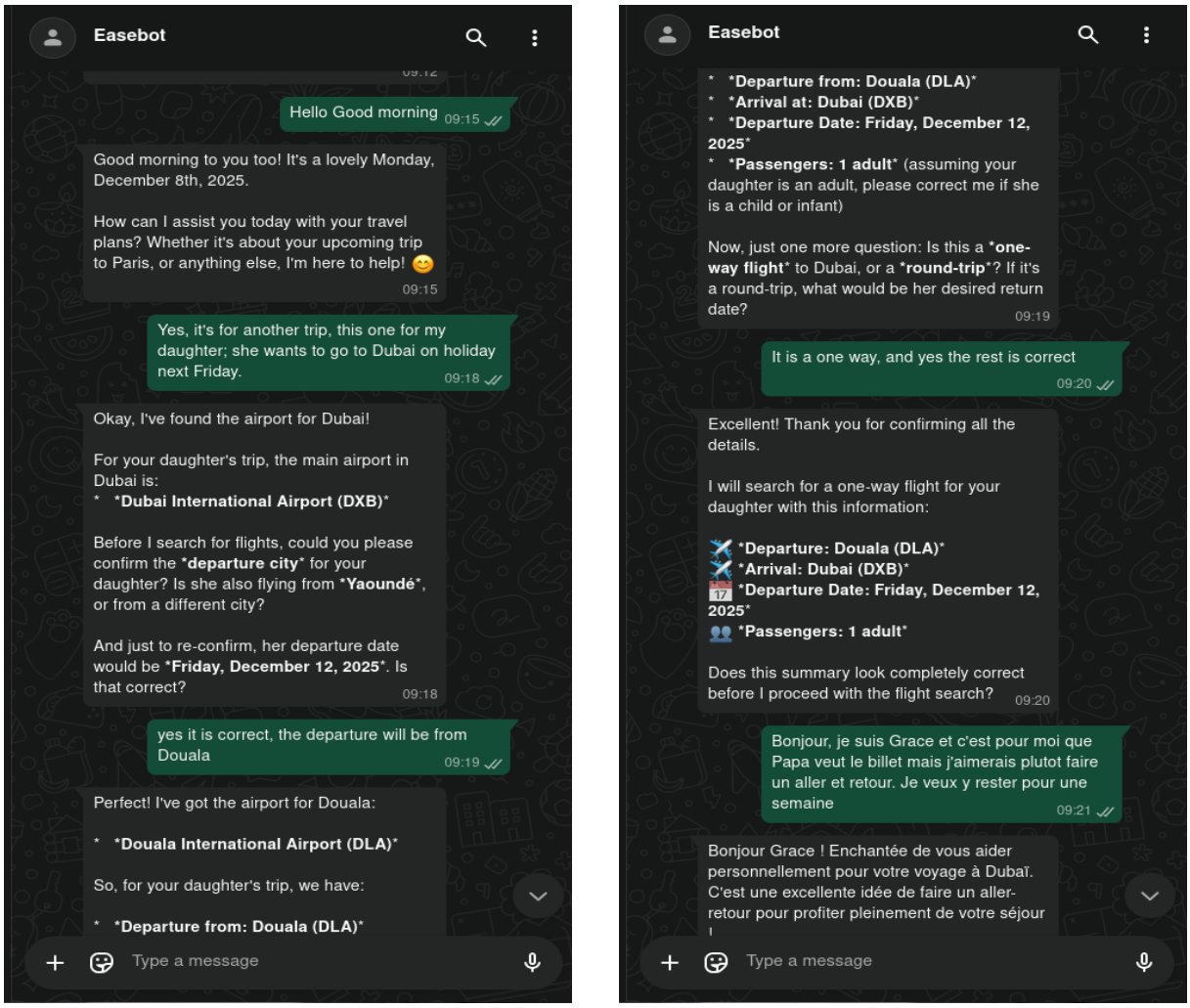

Figure 4: **Dialogue State Management:** Example showing the assistant's handling of complex conversational dynamics. The system manages a context shift (father to daughter), corrects initial assumptions (one-way to round-trip), confirms multiple parameters through clarification questions, and seamlessly switches languages (English to French) while maintaining contextual coherence.

agency experiencing these challenges and was designed to handle all customer inquiries related to the agency's services, including bookings, pricing, schedules, policies, and general support. This real-world deployment demonstrates the practical applicability of the solution and its potential to improve response times, enhance customer service quality, and reduce operational burden for travel agencies.

Figures 4 and 5 illustrates a typical interaction with the proposed chat-bot. When a user restarts a session, the system retains the context of previous conversations and offers the option to resume where the dialogue left off. After the required information is gathered, the chat-bot initiates a flight search only once the user explicitly confirms that the collected details are correct. In addition, the system seamlessly adapts to the user's language primarily (English and French), which are among the most widely spoken languages in many African countries. The response time is also remarkably fast, giving the impression of an almost real-time conversation with a human agent.

The example presented in this paper highlights the chat-bot's ability to remain aligned with the intended domain whether the query concerns flight booking or any other service offered by the company, while delivering a genuinely conversational and context aware user experience.

Conclusion and Future Work

In this work, we present a conversational chat-bot powered by a large language model (LLM), equipped with a Retrieval-Augmented Generation (RAG) system, and designed for a travel agency offering a wide range of services. The virtual as-

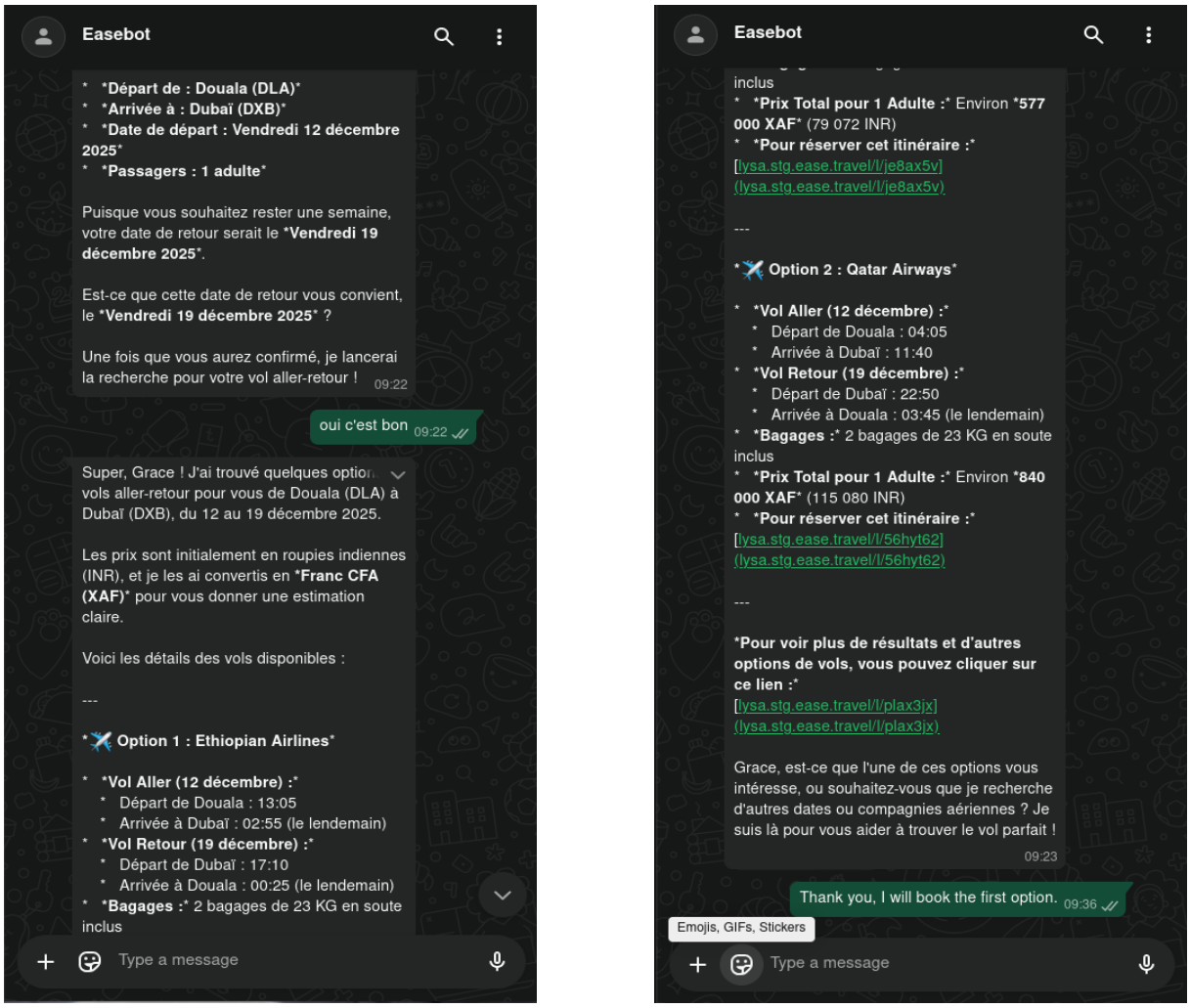

Figure 5: **Example Conversation Flow:** An illustrative WhatsApp exchange demonstrating the assistant's flight booking capability. The system extracts travel parameters (origin, destination, dates), proposes a return date, presents multiple flight options with converted pricing (INR to XAF), and provides booking links.

sistant demonstrates the ability to understand user queries, provide relevant guidance, and support clients throughout the interaction, while maintaining a smooth conversational flow thanks to its low latency and user friendly tone. For future work, we will focus on two main directions. First, we plan to develop a voice-enabled version of the chatbot. This will include integrating speech-to-text and text-to-speech models, enabling users to interact through WhatsApp voice notes. Second,we aim to integrate African languages into the system to increase accessibility and cultural relevance. We will begin with widely spoken languages such as Swahili and Hausa, and gradually expand to others, ensuring that the chatbot becomes more inclusive and better aligned with the linguistic diversity of the continent. Such a feature would significantly improve accessibility, particularly for users with low literacy levels or those who naturally prefer voice communication. By supporting spoken interaction, the chatbot would become more intuitive and adaptable to real-world user behaviors across Africa.

References

David Ifeoluwa Adelani, Graham Neubig, Sebastian Ruder, Shruti Rijhwani, Michael Beukman, Chester Palen-Michel, Constantine Lignos, Jesujoba Alabi, Shamsuddeen H Muhammad, Peter Nabende, and 1 others. 2022. Masakhaner 2.0: Africa-centric transfer learning for named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4488–4508.

Fahd Azaroual. 2024. Artificial intelligence in africa: Challenges and opportunities. *Policy Brief. PB*, pages 23–24.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie

Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2401.XXXXX*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 1. others. 2022. palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

LangChain Community. 2023. Langchain: Building applications with llms. https://docs.langchain.com/oss/python/langchain/.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).

Google DeepMind. 2024a. Gemini 2 flash model card. https://modelcards.withgoogle.com/assets/documents/gemini-2-flash.pdf.

Google DeepMind. 2024b. Gemini flash. https://deepmind.google/technologies/gemini/flash/.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: main volume*, pages 874–880.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, and 1 others. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Stephen King, Toni Morrison, Ernest Hemingway, F Scott Fitzgerald, and Mark Twain. 2025. Digital innovation and automation in african enterprises.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Y Li, J Chen, and X Wang. 2023. Prompt engineering for large language models: A survey. *arXiv preprint*.

Reine Marie Ndéla Marone and Moustapha Mbengue. 2025. Chatbots and artificial intelligence to support digital university libraries in africa: Opportunities and challenges. *Digital Libraries Across Continents*, pages 72–92.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

Michael F McTear. 2002. Spoken dialogue technology: enabling the conversational user interface. *ACM Computing Surveys (CSUR)*, 34(1):90–169.

Wilhelmina Nekoto, Vukosi Marivate, Tshinondiwa Matsila, Timi Fasubaa, Taiwo Fagbohungbe, Solomon Oluwole Akinola, Shamsuddeen Muhammad, Salomon Kabongo Kabenamualu, Salomey Osei, Freshia Sackey, and 1 others. 2020. Participatory research for low-resourced machine translation: A case study in african languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2144–2160.

Keneth Ogueji and 1 others. 2021. Afriberta: African multilingual language model. In *NeurIPS Workshop on AfricanNLP*.

Postman. 2024. What is an api? a beginner's guide to apis. https://www.postman.com/what-is-an-api/.

Model Context Protocol. 2023. What is the model context protocol (mcp)? https://modelcontextprotocol.io/docs/getting-started/intro.

Qdrant Team. 2023. Qdrant: High-performance vector database for scalable ai applications. https://qdrant.tech.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Joseph Weizenbaum. 1966. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.

Jun Xiao, Sheng Zhang, Zhuosheng Zhang, Fei Huang, and Heng Ji. 2024. Bge-m3: Multi-functionality, multi-linguality, and multi-granularity embedding model. https://bge-model.com/bge/bge_m3.html.